Towards a Passive Measurement-based Estimator for the Standard Deviation of the End-to-End Delay

Nghi Nguyen, Thomas Begin, Anthony Busson, Isabelle Guérin-Lassous

Towards a Passive Measurement-based Estimator for the Standard Deviation of the End-to-End Delay

Huu-Nghi Nguyen, Thomas Begin, Anthony Busson and Isabelle Guérin-Lassous
Université Lyon 1, ENS Lyon, Inria, CNRS, UMR 5668 - Lyon, France

***Abstract*— In this paper, we propose an algorithm to estimate the second moment of the end-to-end delay experienced by the packets of a flow based only on delay measurements locally collected by the network nodes. Our solution estimates the standard deviation of the end-to-end delay in an easy and computationally efficient way. Based on thousands of simulations using a real-life trace, our solution is found to be accurate, typically differing by only a few percent from the actual value of the standard deviation of the end-to-end delay.**

1. Introduction

The increasingly complexity of present-day computer networks and the need to cleanse the existing architecture has driven researchers to come up with the new paradigm of Software-Defined Networking (SDN) [1]. In a nutshell, with SDN, the control plane is not executed on the forwarding nodes (routers) but rather managed by a centralized logical controller which is in charge to "program" the network nodes with adequate rules. These rules may pertain to various objectives, ranging from routing and Quality of Service (QoS) to security objectives. To determine these rules, each node periodically communicates to the controller local information as the list of its neighbors, the characteristics of its links as well as parameters regarding the workload activity (typically expressed as counters). Then, it is up to the controller to design network policies that will be translated into rules before being sent to the nodes.

Owing to the controller role and to its awareness of the network topology and activity, a SDN network may be able to implement more efficient and tailored policies than classical IP networks with a distributed control plane. This may lead to a better processing of flows with stringent requirements in terms of QoS (e.g., voice-over-IP, Video-on-Demand, as well as other interactive applications). For instance, it is envisioned that the controller may perform an admission control on incoming flows to ensure that accepted flows experience the right level of QoS. Another example may be the QoS routing in which the controller attempts to find paths across the network that comply with a given level of QoS.

A significant challenge for the SDN controller is to take relevant decisions related to end-to-end performance of a flow such as the end-to end loss ratio, the end-to-end delay, and the available bandwidth while having at its disposal merely local measurements collected on each node. In most cases, the first moment of the end-to-end performance parameters (e.g., the mean value of the end-to-end delay) is straightforward to get. However, to accurately forecast the influence of a new flow entering into the network, or the impact of a workload increase, the mere knowledge of the first moment may be insufficient. For instance, assuming the mean and the standard deviation on the end-to-end delay of a flow are known, and given an apriori knowledge on the form of its probability distribution, it becomes possible to estimate the fraction of packets that will experience an end-to-end delay exceeding a certain value (e.g. quantiles). Also, for many applications, the variability in the end-to-end delay (e.g., jitter) needs to be taken into account in order to forecast the actual QoS obtained by a flow. Hence, the efficiency of the SDN controller may be improved if this latter incorporates algorithms that allow the inference of the first two moments of these end-to-end metrics from the local measurements sent by the nodes. In case the measured quantities at each node can be handled as being independent from each other, this inference does not represent a real issue. But, more sophisticated techniques are required if the measurements from the different nodes are correlated. Broadly speaking, correlations result from the current states of the nodes (e.g., number of packets queued in the buffer) that process the flows. These correlations are difficult to characterize and they must be evaluated through algorithms that offer a trade-off between the computational and communication complexity and the attained accuracy. In this paper, we focus on the end-to-end delay of a flow, which is a key performance parameter to reflect the QoS experienced by a flow. We propose a solution to estimate the first two moments of the end-to-end delay experienced by the packets of a flow in a wired network based only on delay measurements locally collected by the network nodes.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work. Section 3 describes the scenario under consideration in this paper as well as our proposed algorithm to estimate the mean value and the standard deviation of the end-to-end delay of flows. The accuracy of our proposed solution is handled in Section 4. Section 5 concludes this paper.

2. Problem statement and related works

As discussed above, our goal in this paper is to present algorithms that allow the SDN controller to accurately estimate, within its domain, the end-to-end delay of flows

. This work is funded by the French National Research Agency ANR INFRA DISCO under the "ANR-13-INFR-013" project.

through their first two moments. We assume that measured samples of end-to-end delays are unavailable. Many reasons may hinder the measurement of the end-to-end delay: (i) the difficulty to estimate the clock skew between the destination and the source [2]–[4], (ii) the hazard of relying on a single server to communicate the clock because of the latency to transfer syncing NTP (Network Time Protocol) packets [3], (iii) the hazard of relying on several distributed servers because of the clock-drifting phenomena [4], (iv) the requirement for specialized and expensive timing equipments embedding GPS capability [5]–[8], (v) the use of probing packets which may affect the perfomance of existing and regular traffic [4], (vi) the impact of the probing packets profile (e.g. probing packet size, sending rate) on the measurements [8], (vii) the requirement of differentiating probing packets at their departure and then recognizing them at their arrival, (viii) the need of a long enough measurement interval duration (e.g., 10 minutes) to get meaningful statistics [7] (ix) the introduction of errors and uncertainties with any specific measurement method [2], [4], [5], (x) the impact of some sampling methods on the measurement quality [4].

Hence, instead of directly measuring the end-to-end delay using probing or data packets, we rely on the development of algorithms to infer its values based on available measurements, namely data collected on the forwarding nodes.

In addition to forwarding packets to the corresponding interfaces, the nodes also collect measurements on their current state. Typically these measurements are expressed as counters and they track the number of bytes or of packets that have been processed by the node during a given time period. Other possibly measured quantities include the size of incoming packets, the queue length in the buffer of the node as well as the queueing delay spent by each packet waiting for its transmission in the buffer [9]. Assuming samples of these latter quantities can be obtained, it is then straightforward to compute estimated values for their mean as well as for their variance (or equivalently standard deviation) based on the classical empirical estimators.

Unlike the mean value for the end-to-end delay, which can be easily derived from the local measurements, the case for the standard deviation is much more complex. A first and trivial solution could consist to approximate the standard deviation of the end-to-end delay by equalling its variance as the sum of the individual variances observed at each node. Unfortunately, the accuracy of such an approach is typically poor. This inaccuracy results from neglecting fundamental aspects in the computation of the standard deviation, namely the covariance terms, which matter for the spatial correlations (see Section 4).

3. System considered and its approximate solution

In this section we present our proposed approximate solution to estimate the expected value and the variance of the end-to-end delay of flows in a network, without measuring samples of end-to-end delays.

From a flow perspective, the route taken by its packets across a wired network follows, in general, a path of nodes. Hence, we focus our efforts on the case of a network shaped as a path (also called chain) with N nodes, labeled from 1 to N. The link between node i ($i = 1, \ldots, N$) and node $j \neq i$ ($j = 1, \ldots, N$) is characterized by its transmitting capacity (in bps) $\mathcal{C}_{i,j}$ and its propagation delay $\mathcal{R}_{i,j}$ (in sec.) Note that if there is no link between nodes i and j, then we have: $\mathcal{C}_{i,j} = 0$ and $\mathcal{R}_{i,j} = \infty$. At each node, there are as many buffers as out-going network interfaces. The corresponding built-up queues are ruled by a First-Come First-Served discipline. In our case, we assume that buffers are instrumented so that measurements regarding their utilization (e.g., number of queued packets, waiting delay) are available. Therefore, using the classical estimators of the sample mean and of the sample variance [10], each node can compute estimated values for the empirical mean queueing delay as well as for the corresponding variance.

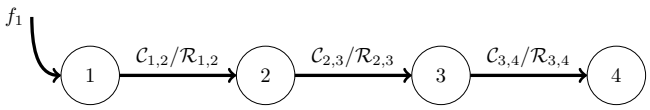


Figure 1: Example of a network with $N = 4$ nodes and one flow f_1.

The workload of a network consists of multiple flows, that originate at different nodes. In this paper, we restrict our analysis to the case of a single flow, which we refer to as f_1. Note that our definition of a flow here refers to a set of packets having the same entry and outgoing points in the network. This flow constituting the network workload can thus be viewed as an aggregation of multiple IP flows, which happen to take the same path on their way to their destination.

We denote by P_{f1} the random variable that characterizes the packet size for the flow f_1 when it enters the network. Based on its routing tables (or flow tables in the case of SDN), the network conveys these packets up to their destination node that is typically several hops away from the source node. Fig. 1 illustrates an example of a network with $N = 4$ nodes and flow f_1.

Along its course towards its destination, a packet is potentially delayed by four types of delays: (i) the processing delays, which mostly consist of handling the packet header and determining its next interface, are typically much less than a microsecond, and hence we neglect them in the computation of the end-to-end delay; (ii) the transmission delays for a packet that depend on its size as well as on the links capacities; (iii) the queueing delays of packets that may greatly vary as they depend on the states of the buffers found upon the packet arrival; (iv) the propagation delays affecting a packet which can be easily known in advance since its value is constant and equal to the sum of the propagation delays of the links composing the taken path.

We denote by $\mathcal{W}_{k,k+1}$, $\mathcal{S}_{k,k+1}$, and $\mathcal{R}_{k,k+1}$ the queueing delay, the transmission delay, and the propagation delay experienced by packets on the link between nodes k and $k+1$, respectively. Clearly, unlike $\mathcal{W}_{k,k+1}$ and $\mathcal{S}_{k,k+1}$, $\mathcal{R}_{k,k+1}$ has a constant value for all packets. Also, note that $\mathcal{R}_{k,k+1}$ can

be known in advance, while $\mathcal{W}_{k,k+1}$ and $\mathcal{S}_{k,k+1}$ can be measured on each node k. Thus, we assume that for $\mathcal{W}_{k,k+1}$ and $\mathcal{S}_{k,k+1}$ can be viewed as random variables.

In the following computations, we derive equations for the end-to-end delay of a packet assuming no packets are lost during their delivery to their destination (due to buffer overflows or link errors). This implies that the distribution of the size of packets for flow f_1, i.e. P_{f1}, keeps constant on all the nodes of the chain.

Using these latter notation, the delay experienced by packets of flow f_1 over the link between nodes k and $k+1$, referred to as $D_{k,k+1}$, can be expressed as:

$$D_{k,k+1} = \mathcal{W}_{k,k+1} + \mathcal{S}_{k,k+1} + \mathcal{R}_{k,k+1}. \quad (1)$$

Let us denote by D the end-to-end delay experienced by packets of flow f_1 over their $N-1$ hops from the source to the destination. We have:

$$D = \sum_{k=1}^{N-1} \mathcal{W}_{k,k+1} + \sum_{k=1}^{N-1} \mathcal{S}_{k,k+1} + \sum_{k=1}^{N-1} \mathcal{R}_{k,k+1}. \quad (2)$$

3.1. Expected value for the end-to-end delay

Using the linear property of the expected value operator, the expected value of end-to-end delay for flow f_1, namely $\mathbb{E}[D]$, can be computed as follows:

$$\mathbb{E}[D] = \sum_{k=1}^{N-1} \Big(\mathbb{E}[\mathcal{W}_{k,k+1}] + \mathbb{E}[\mathcal{S}_{k,k+1}] + \mathcal{R}_{k,k+1} \Big). \quad (3)$$

Evaluating $\mathbb{E}[D]$ using (3) is straightforward. Indeed, the expected value of $\mathcal{W}_{k,k+1}$ can be easily measured on the corresponding interface of node k. As for the transmission delays $\mathcal{S}_{k,k+1}$, they are related to the packet size by the equation:

$$\mathcal{S}_{k,k+1} = \frac{P_{f1}}{\mathcal{C}_{k,k+1}}, \quad (4)$$

so that, we have:

$$\mathbb{E}[\mathcal{S}_{k,k+1}] = \frac{1}{\mathcal{C}_{k,k+1}} \mathbb{E}[P_{f1}]. \quad (5)$$

Thus, combining (5) and (3), we obtain the following expression for $\mathbb{E}[D]$:

$$\mathbb{E}[D] = \sum_{k=1}^{N-1} \mathbb{E}[\mathcal{W}_{k,k+1}] + \left(\frac{1}{\mathcal{C}_{1,2}} + \cdots + \frac{1}{\mathcal{C}_{N-1,N}} \right) \times \mathbb{E}[P_{f1}] + \sum_{k=1}^{N-1} \mathcal{R}_{k,k+1}. \quad (6)$$

3.2. Variance and standard deviation for the end-to-end delay

When it comes to the estimation of the variance for the end-to-end delay of flow f_1, several approaches are possible.

3.2.1. First proposed solution. A first possible way consists to attempt to obtain samples of the whole end-to-end delay. To do that, we resort on a simple procedure, which enables the first node of the path, i.e. node 1, to forecast the queueing delays experienced by a packet at each subsequent node of the path. This computation merely requires node 1 to know, for each packet, its packet size and its inter-arrival times with the previous packet of flow f_1. Let us denote by p^m the m-th packet of the flow ($m > 1$), and by $\delta^m_{i,i+1}$ ($m > 1$) the time that elapses between the arrival at node i ($i = 1, \ldots, N-1$) of packet p^{m-1} and that of p^m (both being sent to node $i+1$ for their next hop). We designate by $w^m_{i,i+1}$ and $s^m_{i,i+1}$ the respective values for the queueing delay and of the transmission delay experienced by the m-th packet on the link between nodes i and $i+1$ ($i = 1, \ldots, N-1$). Given the First-Come First-Served discipline of buffers at each node, it follows that for any m-th packet of flow f_1:

$$\begin{cases} w^1_{i,i+1} &= \max\{0, w^0_{i,i+1} + s^0_{i,i+1} - \delta^1_{i,i+1}\} \\ \delta^m_{i+1,i+2} &= s^m_{i,i+1} + \max\{0, \delta^m_{i,i+1} - w^{m-1}_{i,i+1} - s^{m-1}_{i,i+1}\} \end{cases}, m > 1. \quad (7)$$

If node 1 applies Eq. (7) on every incoming packet of flow f_1, it can then forecast values for their queueing delays at subsequent nodes, as well as for its end-to-end delay up to the destination node. Said differently, node 1 can obtain (predicted) samples for the end-to-end delay of flow f_1. We describe the exact procedure corresponding to the prediction of the end-to-end delay for the m-th packet in Algorithm 1. Finally, based on the classical estimator for the sample variance, node 1 can estimate the variance of the end-to-end delay, and, by taking its square root value, its standard deviation, i.e. σ.

Algorithm 1 First solution proposed for the computation of end-to-end delays

Inputs:
$\delta^m_{1,2}$ ▷ Inter-arrival time for the m-th packet at node 1.
p^{m-1}, p^m ▷ Sizes of the m-th and (m-1)-th packets.
Output:
D^m ▷ End-to-end delay for the m-th packet.
Computation for the m-th packet:
if $m = 1$ **then**
  $\forall j,\ w^1_{j,j+1} \leftarrow 0$
  $\forall j,\ s^1_{j,j+1} \leftarrow p^1 / \mathcal{C}_{j,j+1}$
  $D^1 \leftarrow \sum_j s^1_{j,j+1}$
else
  $D^m \leftarrow 0$
  for j from 1 to $N-1$ **do**
    $s^{m-1}_{j,j+1} \leftarrow p^{m-1} / \mathcal{C}_{j,j+1}$
    $s^m_{j,j+1} \leftarrow p^m / \mathcal{C}_{j,j+1}$
    Recurrent function:
    $w^m_{j,j+1} \leftarrow \max\{0, w^{m-1}_{j,j+1} + s^{m-1}_{j,j+1} - \delta^m_{j,j+1}\}$
    $\delta^m_{j+1,j+2} \leftarrow s^m_{j,j+1} + \max\{0, \delta^m_{j,j+1} - w^{m-1}_{j,j+1} - s^{m-1}_{j,j+1}\}$
    Update D^m:
    $D^m \leftarrow D^m + w^m_{j,j+1} + s^m_{j,j+1} + \mathcal{R}_{j,j+1}$
  end for
end if

Note that Eq. (7) assumes no packet losses (due to buffer overflows), which impacts the accuracy of the proposed estimation, as we will show in Section 4.

3.2.2. Second proposed solution. We now introduce a better solution to estimate with more accuracy the standard deviation of the end-to-end delay in case of packet losses. To do this, we split the variance term into several sub-terms, and evaluate them separately.

Based on (2), this variance verifies:

$$\mathbb{V}[D] = \mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{W}_{k,k+1}\right] + \mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{S}_{k,k+1}\right] + 2 \times \sum_{1 \le i,j \le N-1} \mathrm{Cov}\left(\mathcal{W}_{i,i+1}, \mathcal{S}_{j,j+1}\right). \quad (8)$$

Note that the propagation delays $\mathcal{R}_{k,k+1}$ do not appear in (8) since these latter are constants.

The variance of transmission delays $\mathbb{V}[\sum_k \mathcal{S}_{k,k+1}]$. Based on (4), the total time spent by packets being in transmission along their path towards the destination node (over all the transmitting nodes) can be written as:

$$\sum_{k=1}^{N-1} \mathcal{S}_{k,k+1} = \left(\frac{1}{\mathcal{C}_{1,2}} + \cdots + \frac{1}{\mathcal{C}_{N-1,N}}\right) \times P_{f1}. \quad (9)$$

And because the links capacities are constant, it follows that:

$$\mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{S}_{k,k+1}\right] = \left(\frac{1}{\mathcal{C}_{1,2}} + \cdots + \frac{1}{\mathcal{C}_{N-1,N}}\right)^2 \times \mathbb{V}[P_{f1}]. \quad (10)$$

Eq. (10) shows that the variance of the transmission delays can be easily derived given the variance of the packet sizes.

The covariance between queueing delays and transmission delays $\mathrm{Cov}(\mathcal{W}_{i,i+1}, \mathcal{S}_{j,j+1})$. Intuitively, this covariance term should reflect that, in general, large packets experience statistically different queueing delays than small packets. As is known, the covariance term can be reformulated as:

$$\mathrm{Cov}\left(\mathcal{W}_{i,i+1}, \mathcal{S}_{j,j+1}\right) = \mathbb{E}[\mathcal{W}_{i,i+1}\mathcal{S}_{j,j+1}] - \mathbb{E}[\mathcal{W}_{i,i+1}]\,\mathbb{E}[\mathcal{S}_{j,j+1}].$$

Besides, using (4), $\mathbb{E}[\mathcal{W}_{i,i+1}\mathcal{S}_{j,j+1}]$ can be rewritten as:

$$\mathbb{E}[\mathcal{W}_{i,i+1}\mathcal{S}_{j,j+1}] = \frac{1}{\mathcal{C}_{j,j+1}} \mathbb{E}[\mathcal{W}_{i,i+1}P_{f1}]. \quad (11)$$

This latter term can be evaluated using the measurements collected on each node. Thus, it follows that:

$$\mathrm{Cov}\left(\mathcal{W}_{i,i+1}, \mathcal{S}_{j,j+1}\right) = \frac{1}{\mathcal{C}_{j,j+1}} \Big(\mathbb{E}[\mathcal{W}_{i,i+1}P_{f1}] - \mathbb{E}[\mathcal{W}_{i,i+1}]\,\mathbb{E}[P_{f1}]\Big). \quad (12)$$

Eq. (12) allows us to estimate the covariance between the queueing delays of the packets and their transmission delays.

The variance of queueing delays $\mathbb{V}[\sum_k \mathcal{W}_{k,k+1}]$. To start our computation of $\mathbb{V}[\sum_k \mathcal{W}_{k,k+1}]$, we resort again on the well-known relation for variance, which states:

$$\mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{W}_{k,k+1}\right] = \sum_{k=1}^{N-1} \mathbb{V}[\mathcal{W}_{k,k+1}] + 2 \sum_{1 \le i < j \le N-1} \mathrm{Cov}\left(\mathcal{W}_{i,i+1}, \mathcal{W}_{j,j+1}\right). \quad (13)$$

In (13), only the covariance term is left unknown since each node can provide estimations for the variance of the queueing delays using its measurements. We now explain how we estimate the value of this covariance term. By definition, we have:

$$\mathrm{Cov}\left(\mathcal{W}_{i,i+1}, \mathcal{W}_{j,j+1}\right) = \mathbb{E}[\mathcal{W}_{i,i+1}\mathcal{W}_{j,j+1}] - \mathbb{E}[\mathcal{W}_{i,i+1}]\,\mathbb{E}[\mathcal{W}_{j,j+1}]. \quad (14)$$

Of course, the difficulty in Eq. (14) stems from the need to evaluate the expected value of the product of two quantities, i.e., $\mathcal{W}_{i,i+1}\mathcal{W}_{j,j+1}$, that occur at different nodes. Therefore, we cannot derive the sought value based on the measurements collected on each node. However, using Eq. (7), any node i can forecast the queueing delay for the m-th packet at any subsequent node j $(j > i)$ in the path. We denote by $\mathcal{W}_{i|j,j+1}$ the random variable that represents the forecasted values by node i for the queueing delays at node j. Of course, we have, in general: $\mathcal{W}_{j,j+1} \neq \mathcal{W}_{i|j,j+1}$ since the latter is based on Eq. (7) and so does not take into account the potential packet losses. Nonetheless, the covariance term in Eq. (14) can be approximated by replacing $\mathcal{W}_{i,i+1}\mathcal{W}_{j,j+1}$ with $\mathcal{W}_{i,i+1}\mathcal{W}_{i|j,j+1}$. Hence, we can estimate the covariance terms in Eq. (13), which in turn, allows us to get an approximate values for the variance of the end-to-end delays based on Eq. (8).

Table 1: Distribution of the packet size in the real-life trace.

Packet sizes (bytes)	< 150	150 - 1400	> 1400
Frequency (%)	45.48	24.22	30.30

4. Performance evaluation

4.1. Description of the simulations

To assess the accuracy of our proposed solutions , we run several thousands of experiments using the discrete-event simulator NS-3 [11]. NS-3 is a common tool to simulate network systems, which allows to generate various scenarios through a set of built-in classes that implement network components from the application to the physical layers.

We consider a scenario with a network shaped as a path (i.e. chain) with $N = 4$ nodes and one flow as shown in Fig 1. The link capacities are $\mathcal{C}_{1,2} = 8$, $\mathcal{C}_{2,3} = 5$, $\mathcal{C}_{3,4} = 3$ (Mbps). Hence, the capacity of this path, which is ruled by the link having the smallest transmitting data rate, referred to as the bottleneck link, is 3 Mbps. The propagation delay for all links is set to 2.10^{-3}, i.e. $\mathcal{R}_{1,2} = \mathcal{R}_{2,3} = \mathcal{R}_{3,4} = 2.10^{-3}$ (sec). The size of buffers at each node are set to $15,000$ bytes. In order to consider realistic traffic, the inter-arrival times and the size of the packets at node 1 are derived from a real network trace. The trace was captured at one of the two dormitory network of the university of Stuttgart (Germany) [12]. The trace file corresponds to 4 hours (from 6 to 10 P.M) of transmissions. Table 1 reports the almost bimodal empirical distribution of the packet size obtained with this trace. Indeed, the size of most of packets are close to the maximum, viz. 1500 bytes, or minimum, viz. 64 bytes, values. In order to consider various levels of workload, we alter the actual level of the workload of each excerpt by scaling up or down the times between packets arrivals by a constant multiplying factor.

We evaluate the accuracy of our two proposed estimators for the standard deviation of the end-to-end delay by computing their delivered values on many small excerpts of the trace. Each excerpt has a length 100 packets and permits in turn to get 100 sampled values of delay at each node of the path, which are then used by our proposed solutions to estimate the standard deviation of the end-to-end delay. Overall, we consider 1440 different excerpts.

In all forthcoming figures, our two proposed estimators, which were described in Sections 3.2.1 and 3.2.2, are denoted by $\sigma_{\text{Approx-1}}$ and $\sigma_{\text{Approx-2}}$, respectively.

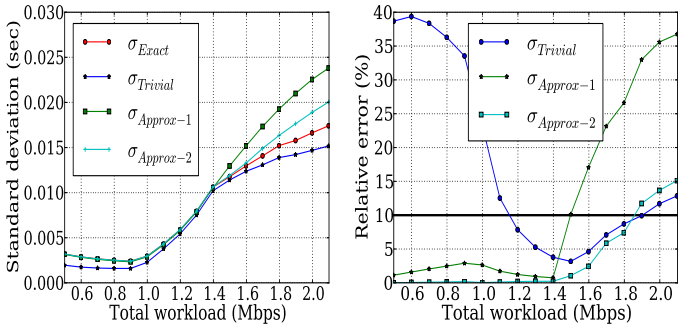
For the sake of comparison, we also implement an *oracle* as well as a *trivial* solution. The former is able to discover the actual value of the end-to-end delay for each packet entering the network. This oracle solution, which can be viewed as a simple implementation of a classical estimator for the standard deviation, is easily programmable and measureable in a discrete-event simulator, but not in a real-life network for all the reasons mentioned in Section 2. Let us denote by σ_{Exact} the corresponding found value for the standard deviation of the end-to-delay.

The so-called trivial solution neglects the correlations between the nodes measurements. It simply estimates the standard deviation of the end-to-end delay by summing up the variances of sojourn times (i.e. queueing plus transmission delays) measured at each node. Thus, it approximates $\mathbb{V}[D]$ by $\sum_{k=1}^{M-1} \mathbb{V}\Big[\mathcal{W}_{k,k+1} + \mathcal{S}_{k,k+1}\Big]$. In subsequent figures, we refer to the corresponding found value as σ_{Trivial}.

4.2. Numerical results

Fig. 2 illustrates the results for many levels of workload (obtained by varying values of the scaling factor between inter-arrival times). We notice in Fig. 2a that while both proposed estimators achieve to capture the general trend of σ_{Exact}, our second estimator, $\sigma_{\text{Approx-2}}$, outperforms the first, $\sigma_{\text{Approx-1}}$. This is even clearer in Fig. 2b, where we depict the relative error committed on the standard deviation of the end-to-end delay by each solution. Indeed, the relative error with $\sigma_{\text{Approx-1}}$ is steadily exceeding that of $\sigma_{\text{Approx-2}}$. More precisely, Fig. 2b indicates that by using $\sigma_{\text{Approx-2}}$ the onset of inaccuracies in the estimation of the standard deviation of the end-to-end delay was postponed to larger levels of utilization of the bottleneck link, say around 70% (instead of 50% with $\sigma_{\text{Approx-1}}$). Quantitatively speaking, with $\sigma_{\text{Approx-2}}$, the relative error is kept well below 5% as long as the workload stays under 1.6 Mbps. For larger values of workload, which tend to represent saturated scenarios given the 3 Mbps transmitting capacity of the bottleneck link, the relative error made by $\sigma_{\text{Approx-2}}$ typically lies within 10 and 15%. While our first solution with $\sigma_{\text{Approx-1}}$ becomes quickly inaccurate as soon as packet losses due to buffer overflows occur, our second proposed solution with $\sigma_{\text{Approx-2}}$ manages to handle some packet losses. As opposed to the first solution that simply neglects potential packet losses, our second solution requires that each node computes the local variances for the queueing and for the transmission delays as well as their covariance term. By doing so, it implicitly and approximately accounts for the possible packet losses occurring in the previous nodes of the path. We observe in Fig. 2 that the trivial solution, labelled by σ_{Trivial}, generally leads to poor results with relative error well beyond 10% in most cases.

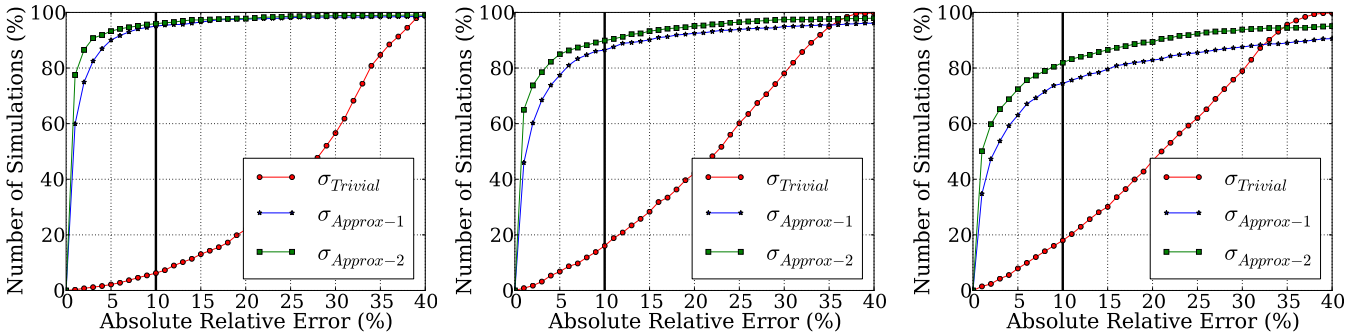
To give a more comprehensive view on the accuracy of our proposed estimators, we apply both of them on thousands (namely, 1440) excerpts of the trace, and we compute for each of them the relative error as compared



(a) (b)

Figure 2: (a) Exact vs Approximate solutions for the standard deviation of the end-to-end delay. (b) Relative error committed by the approximate solutions.

to the value found by σ_{Exact}. We consider three levels of workloads, i.e. 1.0, 1.5 and 2.0 Mbps, corresponding to a lightly, moderately and highly loaded scenarios, respectively. Recall that the bottleneck link has a capacity of 3 Mbps. Note that we do not consider larger levels of workload since they correspond to very unlikely scenarios. Indeed, it is well established that backbone networks, as well as access links, tend to be lightly utilized [13]–[17]. More specifically, the average utilization of the links is typically very low (around a few tens of percent) while links utilization is reported to be larger than 50% but less than 75% if a network failure occurs. Fig. 3 reports the cumulative distribution function of the relative error committed by each of our estimators. For the sake of clarity, we represent in Table 2 the fractions of cases found with a relative error smaller than 10%, 20% and larger than 30%, respectively as well as the mean value. For low levels of workload, Fig. 3a and Table 2 show that both proposed solutions are accurate with more than 95% of the considered excerpts leading to a relative error below 10%. The 10% accuracy threshold is indicated by a vertical asymptote. It is worth noting that for such a level of workload, the trivial solution delivers poor results with around 40% of excerpts having an error exceeding 30%. In Fig. 3b, the path is exposed to a moderate workload, i.e. 1.5 Mbps, whose rate is set to half the transmitting capacity of the bottleneck link of the path. The obtained results are similar to the former example, though the accuracy of our two proposed solutions is slightly reduced. Finally, Fig. 3c reports the obtained results for the case of a workload set to 2 Mbps. In this case, the packet losses are more frequent, and then, unsurprisingly, the accuracy of our solutions is diminished. Furthermore, we notice that the deviations between $\sigma_{\text{Approx-1}}$ and $\sigma_{\text{Approx-2}}$ is deepened, which is in line with our previous observation that our first solution is more affected by packet losses than our second proposed solution. We also notice that while our approximations exhibit a much better behavior than the trivial solution, σ_{Trivial} converges to 100% prior to $\sigma_{\text{Approx-1}}$ and $\sigma_{\text{Approx-2}}$. This implies that the unfrequent worst cases of $\sigma_{\text{Approx-1}}$ and $\sigma_{\text{Approx-2}}$ (90% of cases lead to an error less than 20%) are worse than that of σ_{Trivial}. Under such levels of workload, the proportions of excerpts leading to a relative error less than 10% is above



(a) Workload set to 1 Mbps (33% of bottleneck)

(b) Workload set to 1.5 Mbps (50% of bottleneck)

(c) Workload set to 2 Mbps (66% of bottleneck)

Figure 3: Cumulative distribution function of the relative error committed by the three estimators: $\sigma_{\text{Approx-1}}$, $\sigma_{\text{Approx-2}}$, σ_{Trivial}.

Table 2: Overall distribution of the relative error on the standard deviation of the end-to-end delay.

	Workload set to 1 Mbps				Workload set to 1.5 Mbps				Workload set to 2 Mbps			
Error	≤ 10%	≤ 20%	> 30%	Mean	≤ 10%	≤ 20%	> 30%	Mean	≤ 10%	≤ 20%	> 30%	Mean
$\sigma_{Trivial}$	**6.**25	**21.**94	**43.**40	**26.**42	**16.**04	**42.**08	**21.**94	**21.**18	**17.**99	**46.**67	**21.**11	**20.**47
$\sigma_{Approx-1}$	**95.**07	**97.**57	**1.**67	**3.**05	**86.**46	**92.**50	**5.**07	**6.**56	**74.**44	**82.**85	**12.**43	**12.**22
$\sigma_{Approx-2}$	**95.**97	**97.**78	**1.**11	**1.**90	**89.**86	**95.**14	**2.**57	**4.**21	**81.**94	**89.**37	**6.**18	**7.**76

80% with $\sigma_{\text{Approx-2}}$, while it is around 75% with $\sigma_{\text{Approx-1}}$ and less than 20% with σ_{Trivial} (see Table 2). Finally, Table 2 indicates that the mean error on the standard deviation of the end-to-end delay of the trivial solution tends to oscillate between 20 et 25%, while that of $\sigma_{\text{Approx-1}}$ and $\sigma_{\text{Approx-2}}$ lies between 3 and 13%, and 2 and 8%, respectively.

We have considered lower and larger sizes of buffers. The corresponding results, not shown in this paper due to the lack of space, typically lead to a similar level of accuracy.

5. Conclusion

In this paper, we propose a simple algorithm to estimate the first two moments of the end-to-end delay experienced by the packets of a flow in a wired network based only on delay measurements locally collected by the network nodes. In its current version, our algorithm is thought to handle the case of a single flow. Based on thousands of discrete-event simulations using a real-life trace, our solution is found to be accurate, differing by only a few percent from the actual standard deviation value of the end-to-end delay. Future works will be devoted to the extension of our algorithm to cover more general scenarios including multiple competing flows.

References

[1] B. A. A. Nunes *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks." *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, 2014.

[2] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM," IETF, RFC 2679, 1999.

[3] S. B. Moon, "Measurement and Analysis of End-to-end Delay and Loss in the Internet," Ph.D. dissertation, University of Massachusetts at Amherst, 2000.

[4] J. Wang, M. Zhou, and Y. Li, "Survey on the End-to-End internet Delay Measurements," in *HSNMC*, 2004.

[5] F. Georgatos, F. Gruber, D. Karrenberg, and M. Santcroos, "Providing Active Measurements as a Regular Service for ISP's," in *PAM*, 2001.

[6] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot, "Analysis of Point-To-Point Packet Delay In an Operational Network," in *IEEE INFOCOM*, 2004.

[7] B.-Y. Choi, S. Moon, R. Cruz, Z.-L. Zhang, and C. Diot, "Practical Delay Monitoring for ISPs," in *ACM CoNEXT*, 2005.

[8] A. Hernandez and E. Magaña, "One-way Delay Measurement and Characterization," in *IEEE ICNS*, 2007.

[9] K. Papagiannaki, S. B. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network." *IEEE JSAC*, vol. 21, no. 6, 2003.

[10] A. O. Allen, *Probablity, Statistics and Queueing Theory with Computer Science Applications, Second Edition*. Elsevier, 1990.

[11] T. R. Henderson *et al.*, "Network Simulations with the ns-3 Simulator," ACM SIGCOMM Demos, 2008, code available: http://www.nsnam.org/releases/ns-3.1.tar.bz2.

[12] D. Sass, "The dormitory network "Selfnet" of the University of Stuttgart," Oct. 2004, http://www.ikr.uni-stuttgart.de/Content/I2MP/.

[13] O. Andrew, "Networks are Lightly Utilized, and Will Stay that Way," *Review of Network Economics*, 2003.

[14] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, vol. 17, no. 6, 2003.

[15] A. Nucci, N. Taft, C. Barakat, and P. Thiran, "Controlled use of excess backbone bandwidth for providing new services in IP-over-WDM networks," *IEEE JSAC*, 2004.

[16] J. L. Garcia-Dorado, J. A. Hernández, J. Aracil, J. E. L. de Vergara, and S. Lopez-Buedo, "Characterization of the busy-hour traffic of IP networks based on their intrinsic features," *Computer Networks*, vol. 55, no. 9, 2011.

[17] A. Hassidim, M. Segalov, and A. Shaqed, "Network utilization: The flow view," in *IEEE INFOCOM*, 2013.